



POLITECNICO DI TORINO
Repository ISTITUZIONALE

Formal Verification of Device State Chart Models

Original

Formal Verification of Device State Chart Models / Corno F.; Sanaullah M.. - STAMPA. - Proceedings 2011 Seventh International Conference on Intelligent Environments(2011), pp. 66-73. ((Intervento presentato al convegno The 7th International Conference on Intelligent Environments tenutosi a Nottingham (UK) nel 25-28/07/2011.

Availability:

This version is available at: 11583/2402456 since:

Publisher:

IEEE Computer Society

Published

DOI:10.1109/IE.2011.36

Terms of use:

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

IE'11



The 7th International Conference on Intelligent Environments
25-28 July 2011, (workshops on 25-26 July 2011), Nottingham, United Kingdom

Politecnico di Torino
Dipartimento di
Automatica e Informatica
Torino, Italy



e-Lite

<http://elite.polito.it>

Formal Verification of Device State Chart Models

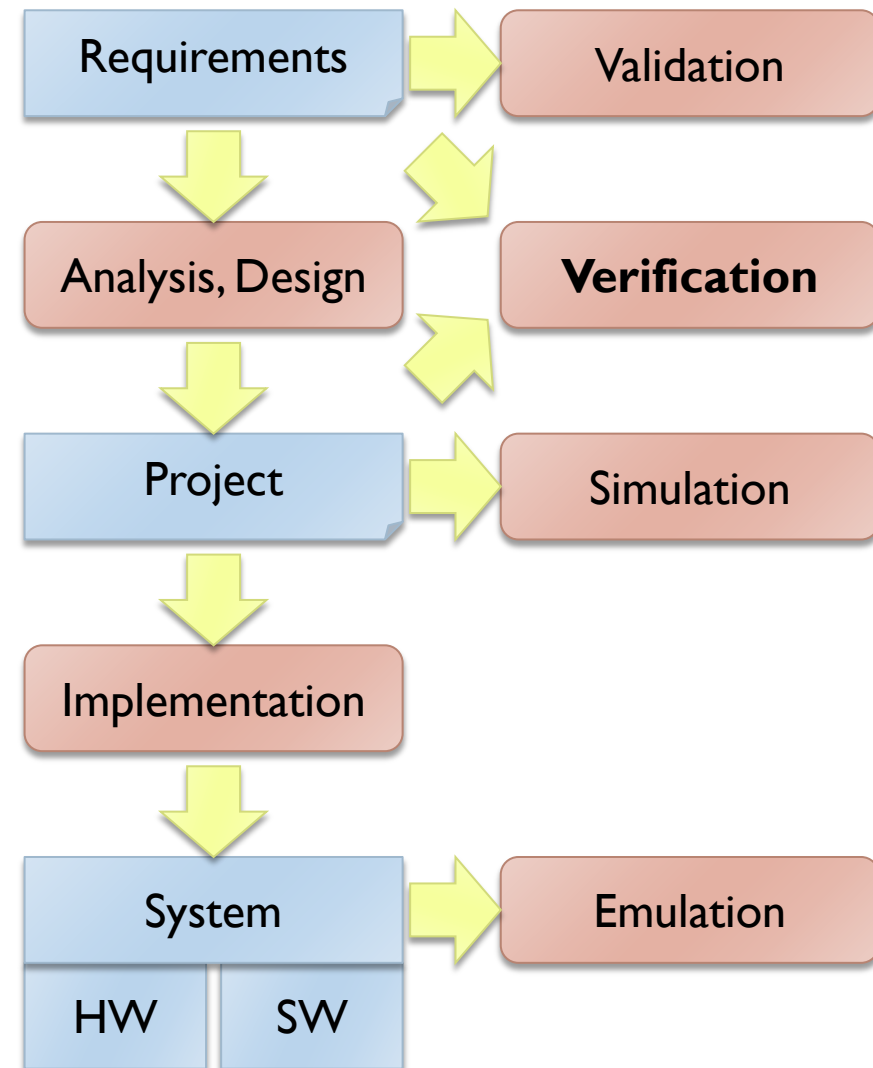
Fulvio Corno, Muhammad Sanaullah

Outline

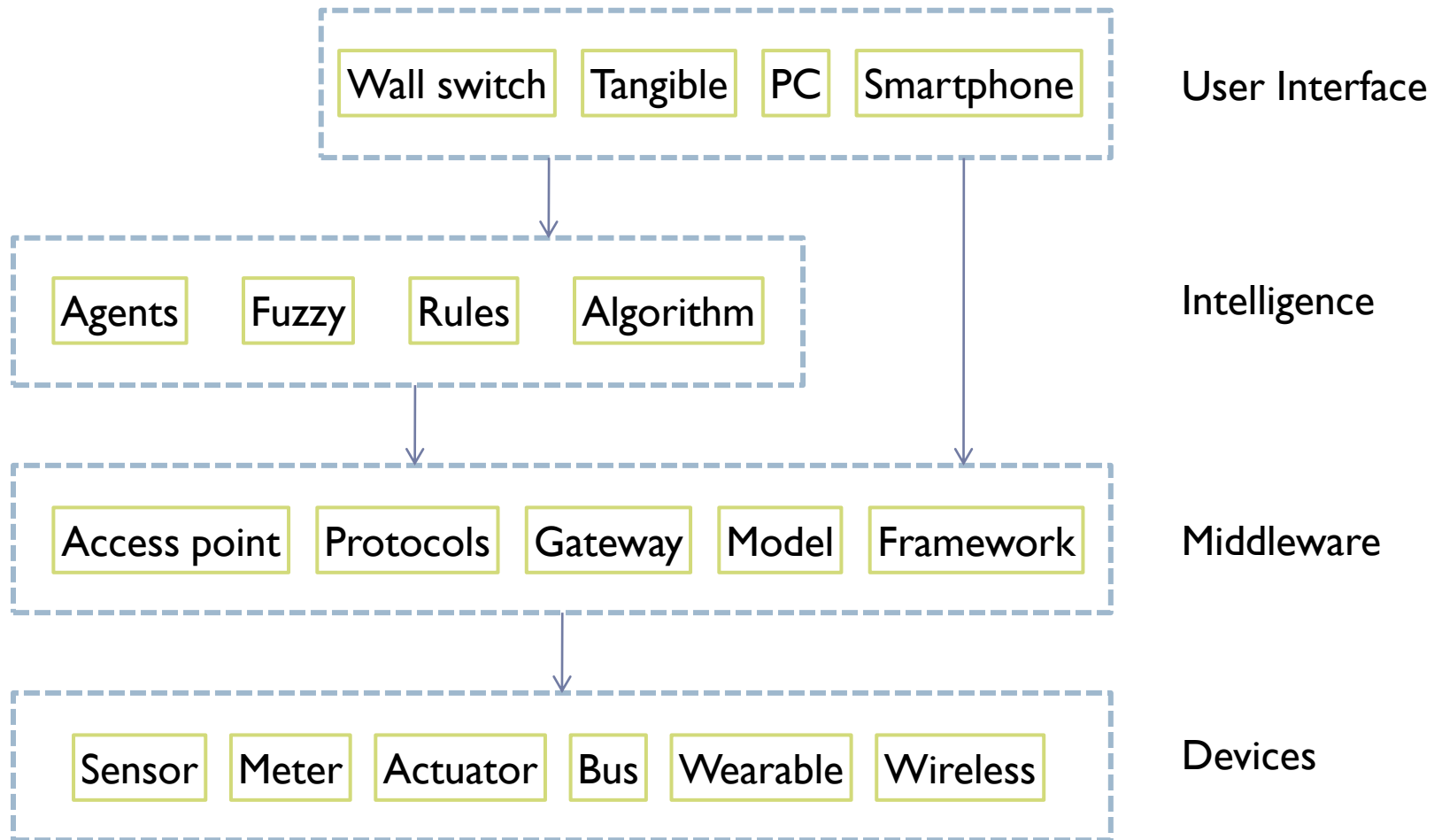
- ▶ Design process
- ▶ Formalisms
- ▶ Verification Methodology
- ▶ Results
- ▶ Conclusions

Design process for IE

- ▶ Intelligent environments gaining acceptance
 - ▶ More installations
 - ▶ Standard solutions
- ▶ Need more structured design process
 - ▶ Less “art”
 - ▶ More “engineering”



Reference model



General Goals

- ▶ Adopt **formal representations** to allow a sound design process
- ▶ Enable validation and verification **throughout** the design process
- ▶ Integrate the solution in the Dog2.1 gateway toolset



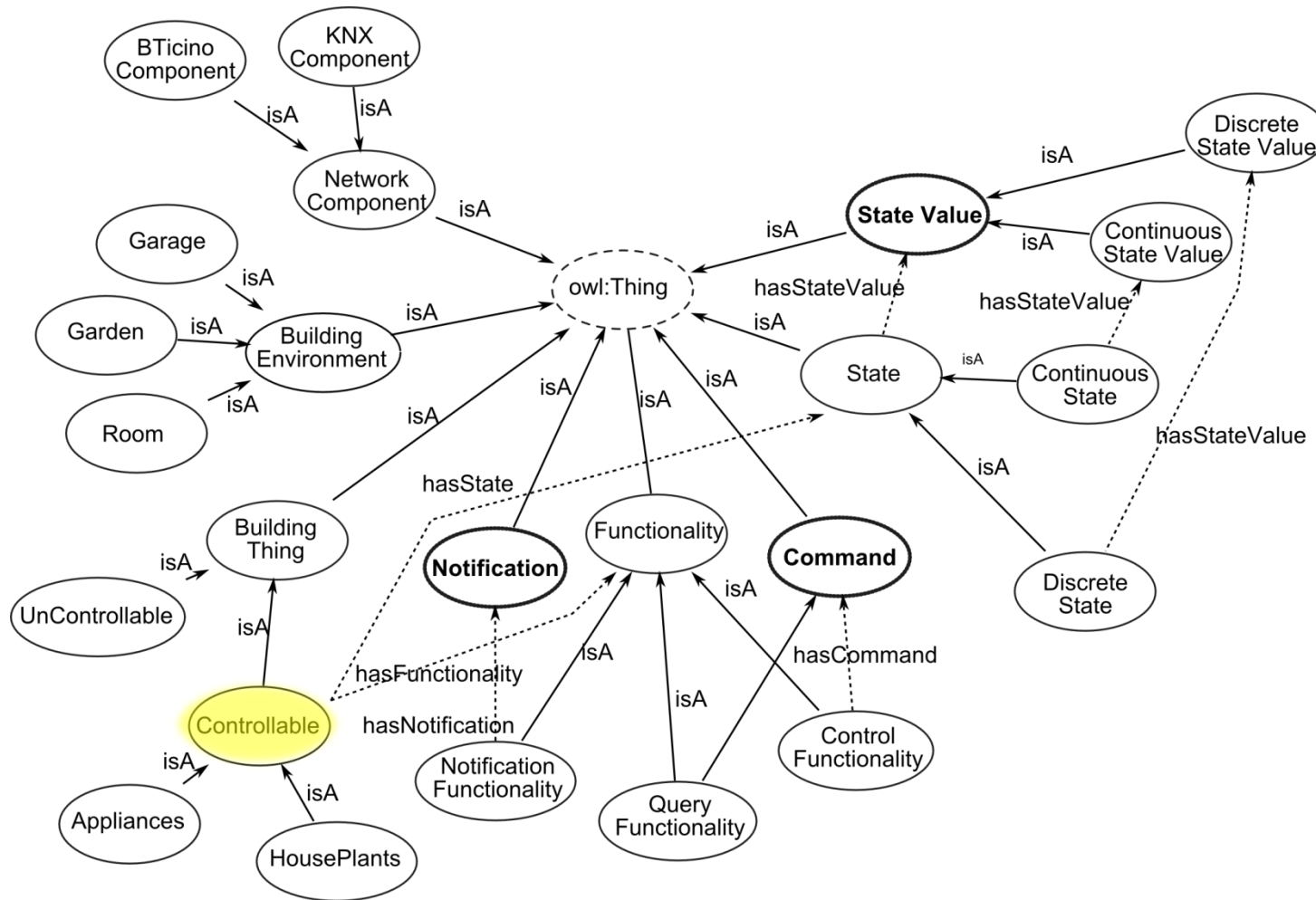
DOG 2.1

<http://domoticdog.sourceforge.net>

Adopted formalisms

Level	Design artifact	Technique	Formalism
User Interface	System requirements	Temporal Logics	UCTL
Intelligence	Intelligent algorithms	State machines	UML Statecharts
Middleware	Device categories	Ontology	DogOnt classes
	System configuration	Ontology	DogOnt instances
Devices	Device models	State machines	UML Statecharts
	Whole system behavior	Parallel state machines	UML Statecharts

The DogOnt ontology



Formalism

UCTL

UML Statecharts

DogOnt classes

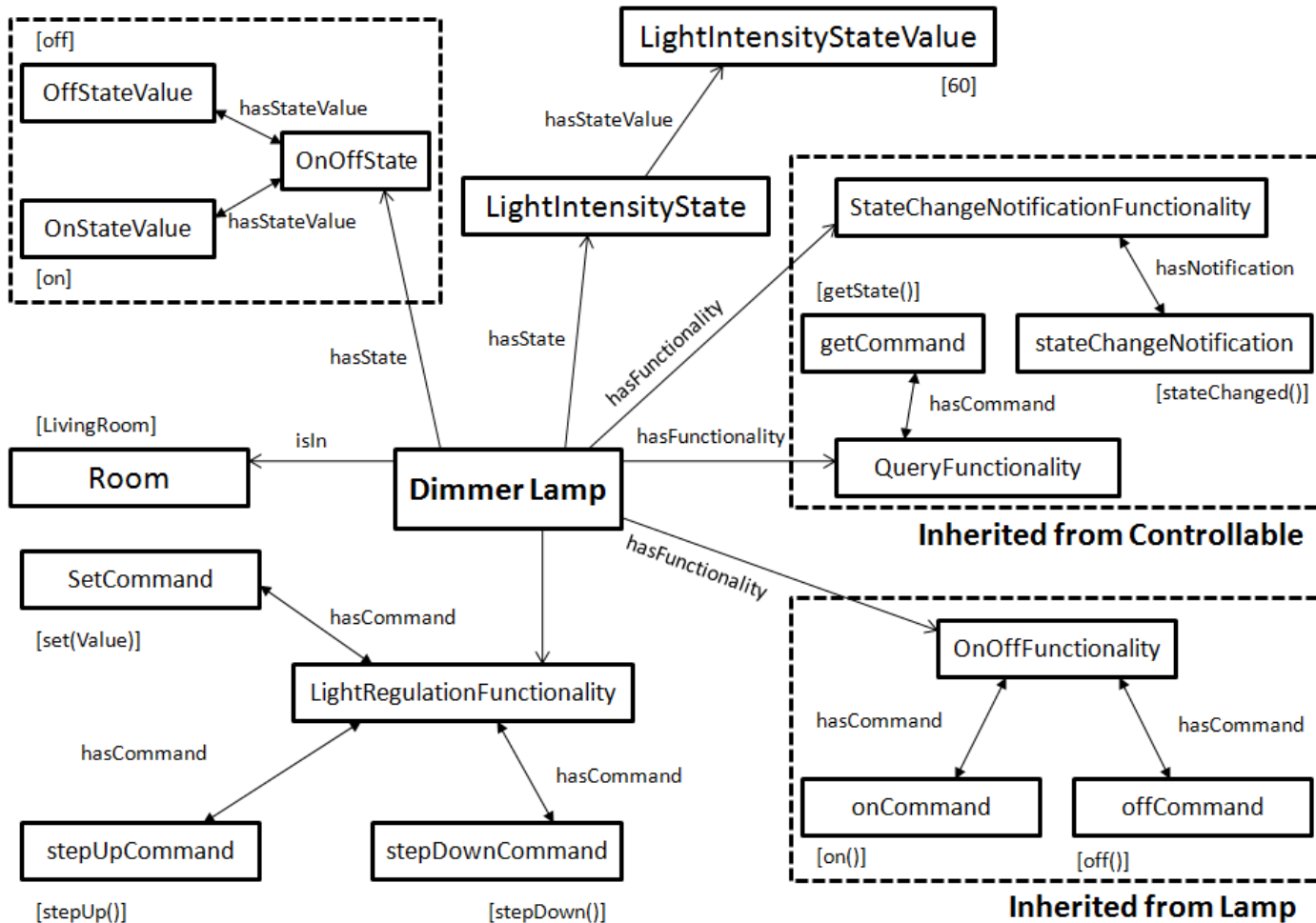
DogOnt instances

UML Statecharts

UML Statecharts

DogOnt instances: DimmerLamp

Inherited from Lamp



Formalism

UCTL

UML Statecharts

DogOnt classes

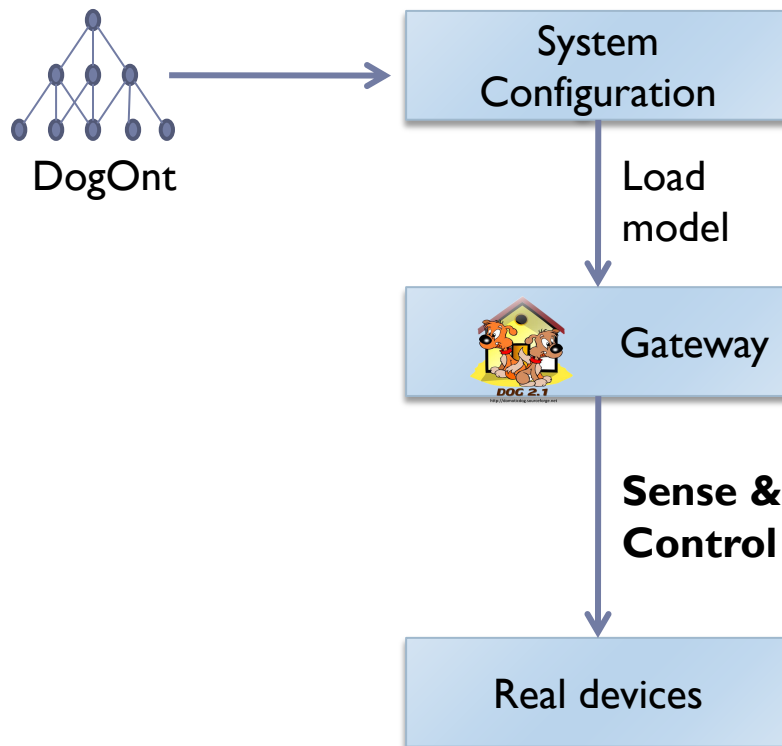
DogOnt instances

UML Statecharts

UML Statecharts

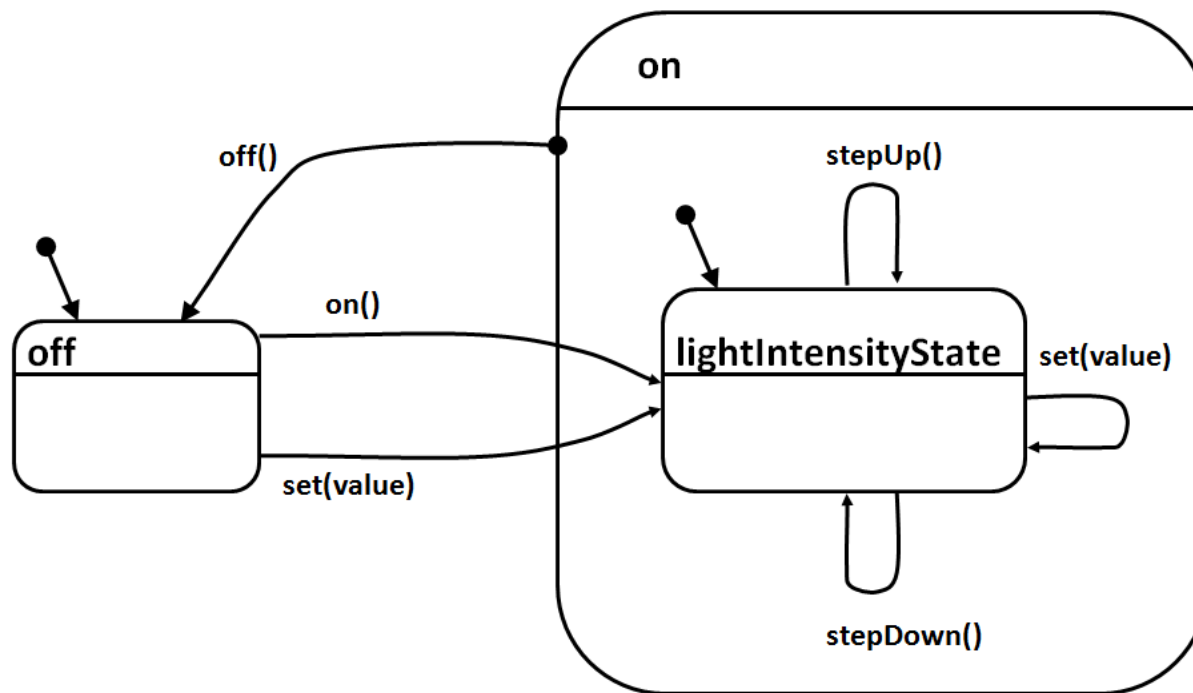
Overall system components

...to be continued...



Device modeling

- ▶ Ontologies are declarative formalisms: device properties
- ▶ For device behavior we need an operational formalism
 - ▶ Statecharts (Harel, 1987, now in UML 2.0)



Formalism

UCTL

UML Statecharts

DogOnt classes

DogOnt instances

UML Statecharts

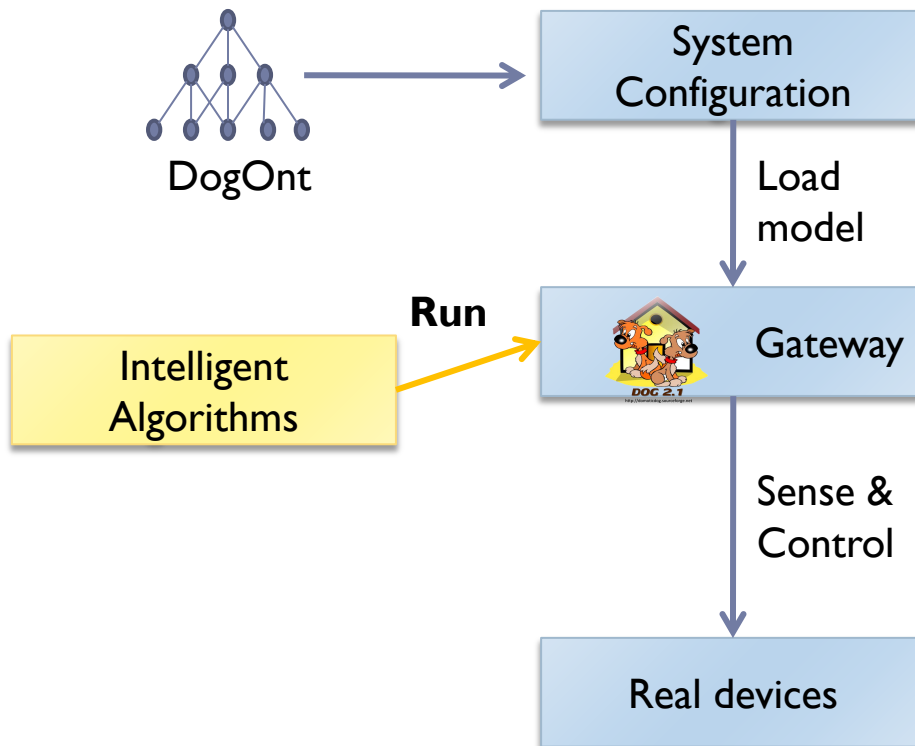
UML Statecharts

Use cases

- ▶ Ontologies are declarative formalisms: device properties
- ▶ For device behavior we need an operational formalism
 - ▶ Statecharts (Harel, 1987, now in UML 2.0)
- ▶ We use Statecharts for
 - ▶ Modeling the behavior of each **device** type
 - ▶ Implementing the **Intelligent Algorithms** within the gateway
 - ▶ Building a **whole-system model** allowing simulation and emulation
- ▶ Statecharts have a formal semantics: formal verification is possible

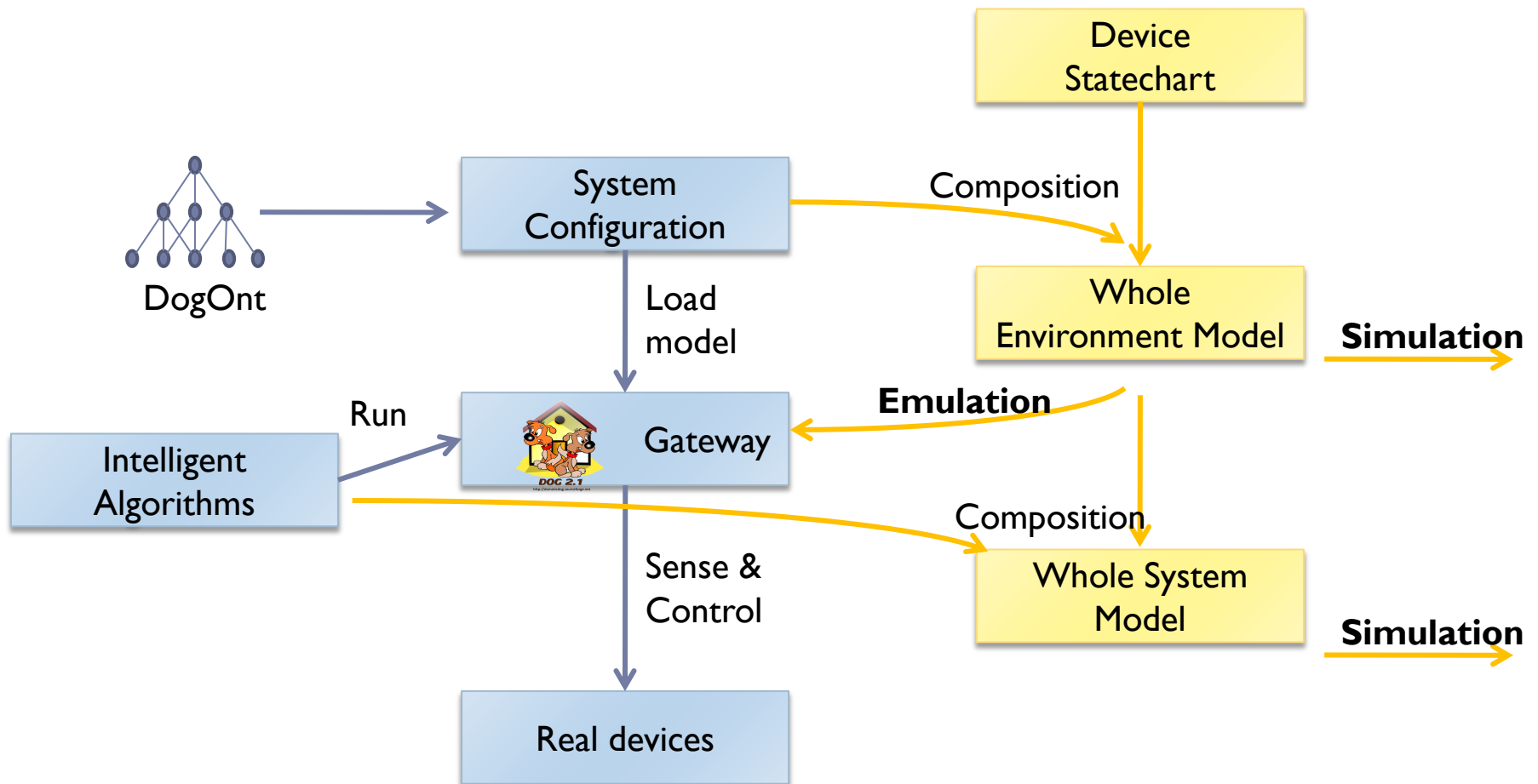
Overall system components

...to be continued...



Overall system components

...to be continued...



Temporal logic

▶ UCTL logic

- ▶ Branching-time
- ▶ State-based and action-based
- ▶ Operators
 - ▶ Next (X,N)
 - ▶ Future (F)
 - ▶ Globally (G)
 - ▶ All (A)
 - ▶ Exists (E)
 - ▶ Until (U)

▶ UMC Model Checker

- ▶ Supports Statecharts as a model

Examples

$$AG \llbracket openRequest(T1) \rrbracket$$

$$A \left[\top \{ \neg openRequest(T1) \} U_{\{ tsDone(T1) \}} \top \right]$$

$$AG \llbracket daDoorOpen(DAExt) \rrbracket$$

$$A \left[\top \{ \neg daDoorOpen(DAInner) \} U_{\{ extDoorClosed() \}} \top \right]$$

Formalism

UCTL

UML Statecharts

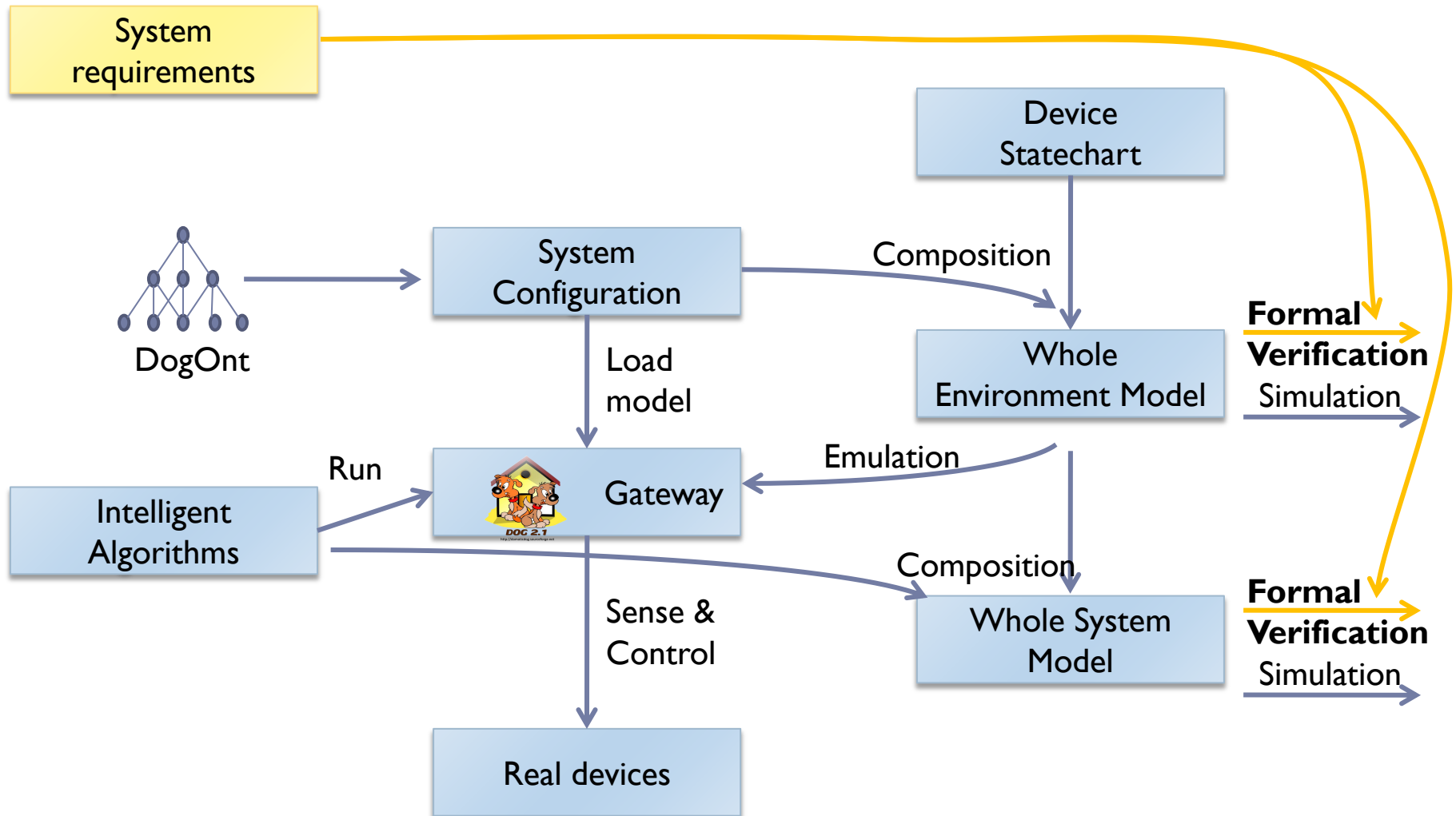
DogOnt classes

DogOnt instances

UML Statecharts

UML Statecharts

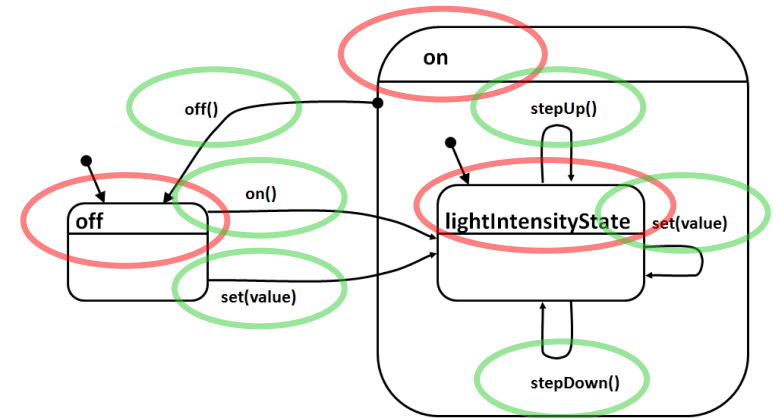
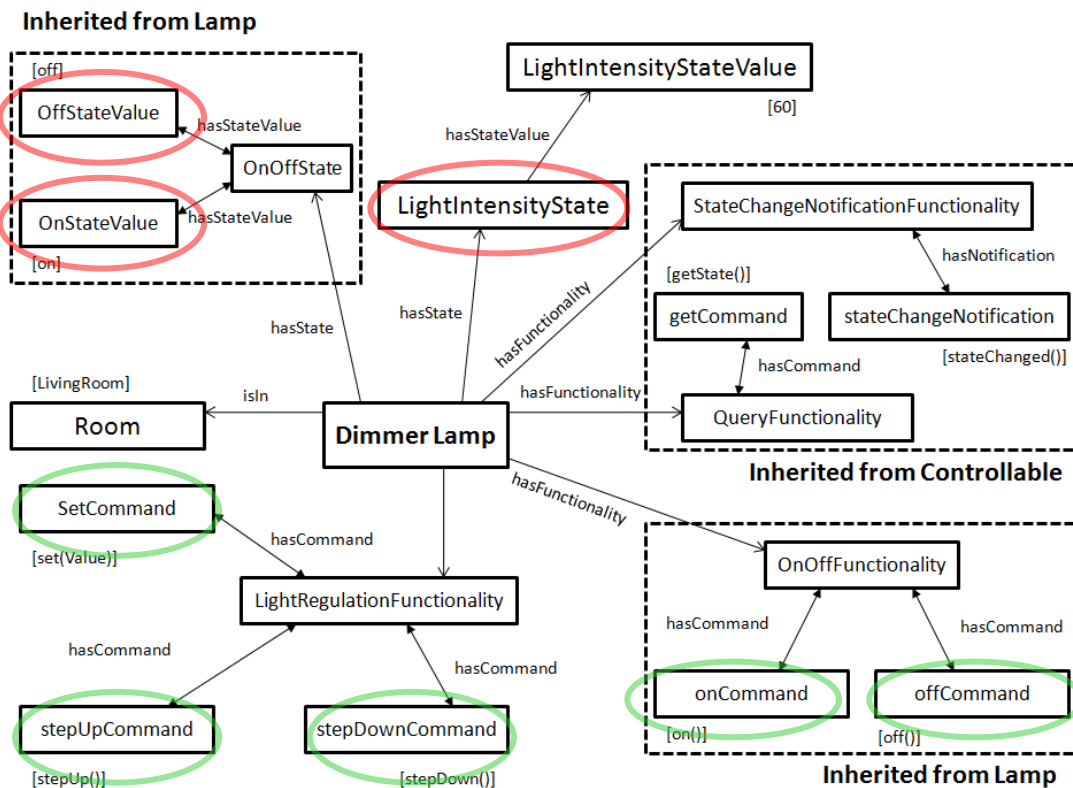
Overall system components



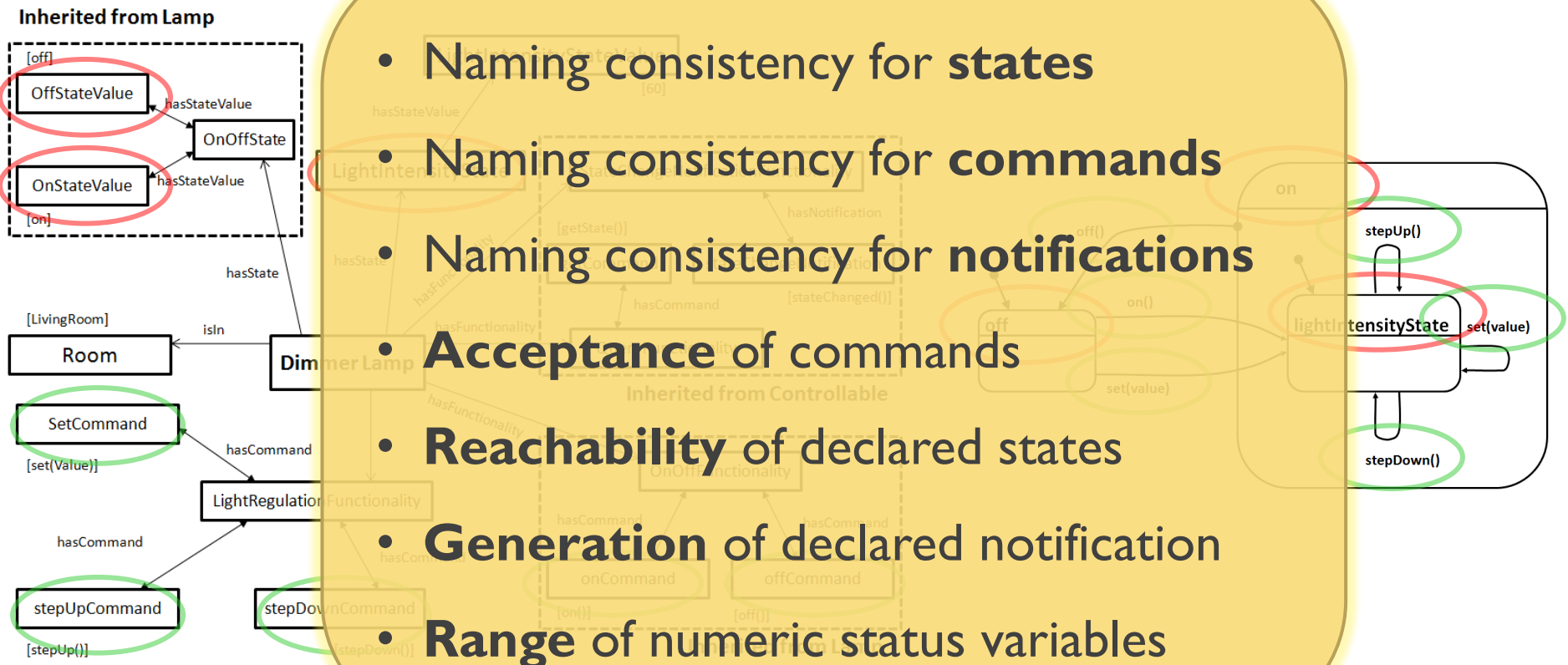
But... (goal of this paper)

- ▶ Formal verification relies on the composition of device state charts
- ▶ Environment control relies on information in DogOnt device properties
- ▶ How to ensure their consistency?
- ▶ Solution: use formal verification, too

The problem

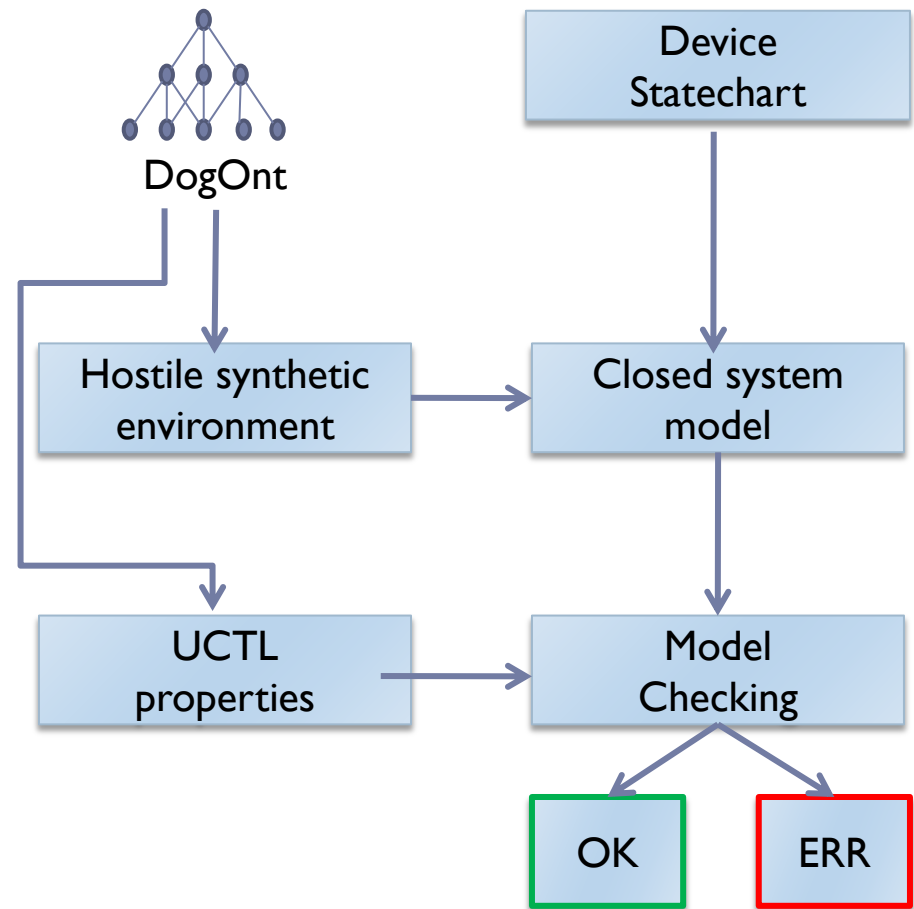


The problem



Approach

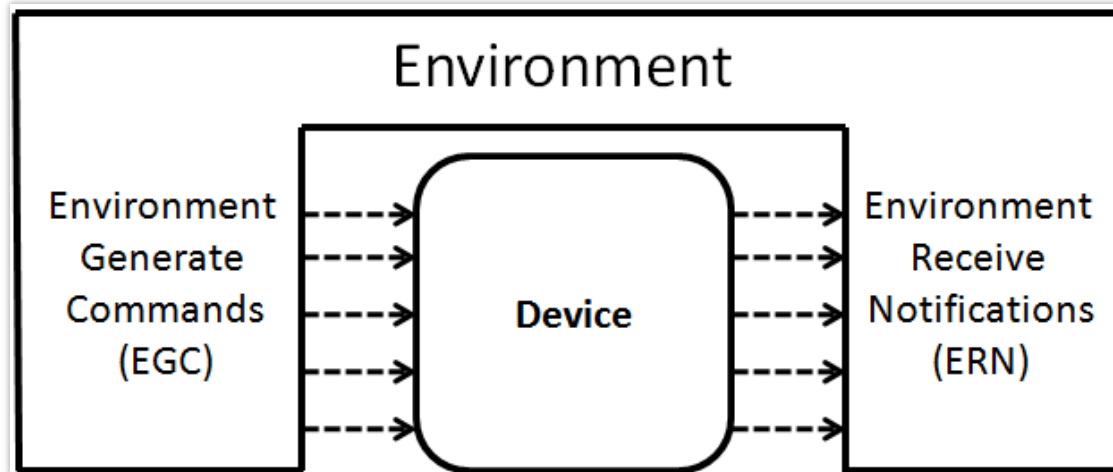
- ▶ From DogOnt, extract UCTL properties
- ▶ From DogOnt, build a synthetic environment for the device
- ▶ Integrate Device State Chart in the synthetic environment
- ▶ For every property
 - ▶ Run Model checker



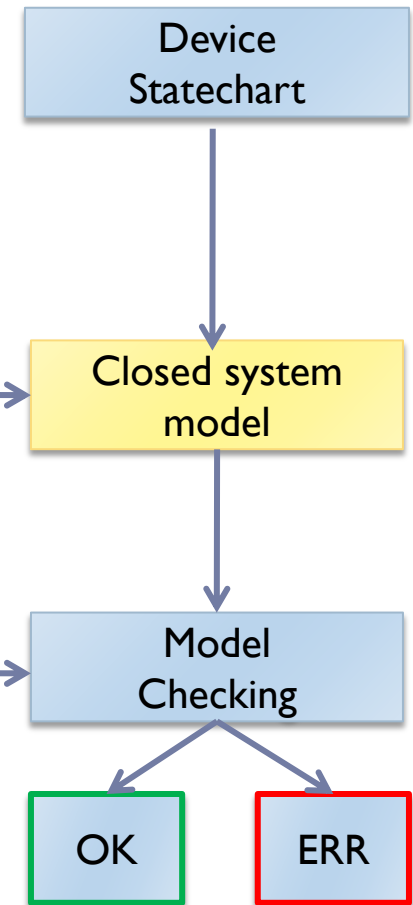
Approach

► From DogOnt extract

Building a closed system model, ready for verification



► Run Model checker



Approach

Example: DimmerLamp generated & verified properties

--Action Properties

--the acceptance of all the commands in DSC

EF {sending(stepDown)} true

EF {sending(stepUp)} true

EF {sending(set)} true

EF {sending(off)} true

EF {sending(on)} true

--

EF {accepting (stepDown)} true

EF {accepting (stepUp)} true

EF {accepting (set)} true

EF {accepting (off)} true

EF {accepting (on)} true

--the generation of all the notifications in DSC

EF {sending(stateChanged)} true

EF {accepting(stateChanged)} true

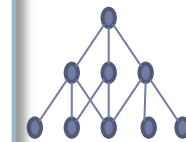
--State Properties

--the reachability of all the states in DSC

EF (offState)

EF (onState)

EF (LightIntensityState)



DogOnt

ostile synthetic
environment

UCTL
properties

Device
Statechart

Closed system
model

Model
Checking

OK

ERR

Experimental Results

- ▶ UCTL Model Checker
- ▶ Dog2.1 standard device classes
- ▶ Device classes verified: 11
- ▶ Number of verifies properties: 114
 - ▶ Some design errors found and corrected
- ▶ CPU time: < 1 sec / property
- ▶ **Formally validated device statechart library in Dog2.1**

Conclusions

- ▶ Engineering the Design Process for Intelligent Environments
- ▶ Formalisms and tools are needed
- ▶ Ontologies, Statecharts, Temporal Logics

Thank you!



<http://elite.polito.it>

<http://domoticdog.sourceforge.net>

fulvio.corno@polito.it